UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,545	03/31/2004	Ram P. Mohan	24858.703.301	7182

23623 7590 03/20/2008

AMIN, TUROCY & CALVIN, LLP
1900 EAST 9TH STREET, NATIONAL CITY CENTER
24TH FLOOR,
CLEVELAND, OH 44114

EXAMINER	
INGBERG, TODD D	
ART UNIT	**PAPER NUMBER**
2193	

NOTIFICATION DATE	DELIVERY MODE
03/20/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
osteuball@thepatentattorneys.com

Office Action Summary	Application No.	Applicant(s)	
	10/815,545	MOHAN ET AL.	
	Examiner	Art Unit	
	Todd Ingberg	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 January 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23 and 27-40* is/are pending in the application.

4a) Of the above claim(s) *24-26* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23,27-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *9/3/2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1 – 23, 27 - 40 have been examined.

Claims 24 – 26 have been canceled.

Claims 1, 9, 16-18, 20, 22 – 23, 36 have been amended.

Request for Continued Examination Processed.

Priority

1. Priority to Abandoned application 09/808,741 with a date of March 14, 2001 has been established. Claim to Domestic priority with 60/189,358 (March 14, 2000) will be reviewed with each response. Claimed subject matter supporting Figures 16 – 52 do not receive the Domestic Priority Date. The effective filing date for claimed subject matter supporting these figures will be determined by the date of the CIP filing.

2. The later-filed application must be an application for a patent for an invention which is also disclosed in the prior application (the parent or original nonprovisional application or provisional application). The disclosure of the invention in the parent application and in the later-filed application must be sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112. See *Transco Products, Inc. v. Performance Contracting, Inc.*, 38 F.3d 551, 32 USPQ2d 1077 (Fed. Cir. 1994).

The disclosure of the prior-filed application, Application No. 60/189, 358, fails to provide adequate support or enablement in the manner provided by the first paragraph of 35 U.S.C. 112 for one or more claims of this application.

3. Claims 1 – 40 fail to be support by the Domestic Priority document. A lack of support to provide enough information for one of ordinary skill in the art to provided the additional claimed

and Specified subject matter. The following is a summary of the unsupported terms: Application logics, Visual node layout, interface block node, interaction node, user interface node, user interface block node, visual representation of a software function, task node interface, execution without compilation of application logic, paused and saved during execution of application logic, restored and resumed saved application logic (including state information), visual node layout, node execution measurements, any automatic generation of documentation, version history abilities, access control of application logic outside normal "accessor functions", (i.e. ANCI C Private, Public), logic for the purpose of modification and multiple access of the application logic for the purpose of viewing, automatically validating application logic against errors, event triggering and conditions.

4. In the event the Applicant elects to argue the denial of the Domestic Priority date. The mapping of the claimed subject matter to the Domestic Document is required to overcome the rejection of priority.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1 – 23, 27 - 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over SoftWire (**SW**) Graphical Programming as taught in USPN #6,425,121 B1 issued July 23, 2002 and filed June 14, 2000 in view of Visual Basic 6 (**VB**) as documented in Rob Thayer's text book Visual Basic Unleashed (September 11, 1998). SoftWire states the invention is an add-on

to Visual Basic (SW, col 2, lines 1-16 and col 6, lines 37 - 53) in further view of Template Software's Workflow system Chapter 9 from 1998 (**WFT**) and in view of USPN # 6,810,401 B1, Thompson et al, issued October 26, 204, filed October 10, 2000 with Provisional priority to October 8, 1999.

Claim 1

SW teaches a method for creating software (SW, Abstract), comprising: providing a plurality of nodes (SW, Figure 4B, #402 and #414) and a directory of applications (SW, Figure 4B, #418, #420, #422), each of an application being created by use of at least a portion of the plurality of the nodes; selecting at least a portion of the plurality of nodes to create a selected node layout that represent a plurality of application logics (SW, Figure 9, shows program in iconic form and SW, col 8, line 55 – col 9 , line 13); and executing the selected node layout by a server program (VB, Server, pages 547 – 549). visually showing the selected node layout as a visual node layout (The combination of SW and VB teach the node layouts (referred to as icons) and WFT teaches node layouts see 8-10), ; monitoring a flow of control through each node in the node layout during execution by showing individual node execution measurement (WFT, both Chapter 8 the Simulator teaches monitoring a flow of control and Chapter 9 WFT Using the Management Status and Error Facilities); and wherein the individual node execution measurements include usage totals , whole execution time and average execution time (WFT, captures both node information and queue information in RSTO Module page 9-50 to 9-55). The intended use of the WFT system is to access RSTO to writing your own statistical routines in addition to the built in routines (page 9-54 to 9-55) . One of ordinary skill in the art would know how to write routines for usage totals, whole execution time and average execution for nodes (page 9-55) as well as for Users, Tasks and Work items. Thompson teaches **employing a solution engine that uses networked objects (Thompson in combination with Template – Template teaches the distributed object environment and Thompson teaches Col 4, lines 30-42)to follow and manage user sessions (See Thompson teaches Figure #27, Users functionality, #402 and #404), wherein the solution engine interfaces with an observation (Thompson, Figure 5, Comment Window), personalization (Thompson, col 8, lines 60-62) and pricing engine as directed by specific nodes (Thompson, col 8, lines 8 – 25 and col 18, lines 41-56)**

SoftWire (SW) teaches the use of icons to perform object based programming. The icons generate Visual Basic code. Visual Basic (VB) is a popular object programming language which enables add-ins/-ons such as SoftWire (VB, Chapter 3 – VB6 Template Manager, page 62) features such as web development and client server architecture. Softwire is an add-on to Visual Basic which teach a rapid development environment. Template teaches the ability to drill down within nodes and view the underlying performance information (WFT, Chapters 8 and 9) of the workflow model both in a simulation (Chapter 8) and the deployed system (Chapter 9). Thompson teaches automated configuration (Thompson, Abstract)Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of

rapid deployment system of SW and VB with the node statistical abilities of Template with the ability to perform automated configuration, because the ability to determine the performance of a workflow systems yields information on how to tune the system and automated configuration yield information on how to tune and calibrates systems based on usage.

Claim 2
The method of claim 1, further comprising: visually displaying the selected node layout as a visual node layout. (SW, col 6, lines 21 – line 53).

Claim 3
The method of claim 1, wherein at least a portion of the plurality of application logics includes a user interaction (SW, Col 6, lines 50 - 54).

Claim 4
The method of claim 3, wherein the user interaction permits a user to interact with the server program (VB, client server, pages 547-549).

Claim 5
The method of claim 3, wherein the user interaction is executable on multiple channels (VB, Web, page 520 and Mail, page 484).

Claim 6
The method of claim 3, wherein the user interaction is executable by at least one of web, voice, e-mail and wireless channels (See the rejection for claim 5).

Claim 7
The method of claim 1, wherein the plurality of nodes includes a user interface node (SW, images and forms, col 1, lines 47 – col 2, line 15).

Claim 8
The method of claim 7, wherein the user interface node includes GUI components and a template for the physical layout of static (SW, Figure 4C the icon is static) and dynamic portions of a user display (SW, Col 17 line 65 to col 18 line 17, properties like caption and attributes can be changed).

Claim 9
The method of claim 8, wherein dynamic portions of the user display are used by the server program at runtime to layout application specific GUI components (see the rejection for claim 7 and VB, page 484, Client – server).

Claim 10
The method of claim 3, wherein the user interaction includes a user interface node, a user interface block node and an interaction node. (SW, figure 4 – nodes for building interaction with

user in GUI and properties SW, col figure 8A right side – Invention is an add-in to VB Properties of VB used in invention).

Claim 11
The method of claim 10, wherein the user interface node and user interface block node (as per claim 10) create a user interaction based on business rules (SW, Figure 9 - illustrate a business rule modeled and VB, page 890 - business logic layer).

Claim 12
The method of claim 11, wherein the interaction node executes the user interaction. See the rejection for claim 10.

Claim 13
The method of claim 1, wherein each node is a visual representation of a software function. (SW, col 31, lines 21 – 42)

Claim 14
The interface of claim 10, wherein each node includes inputs to a software function. (SW, col 31, lines 42 – 67 – data input in example)

Claim 15
The interface of claim 1, wherein the plurality of nodes includes task node interfaces with external components to exchange data information. (SW, col 7, lines 33 – 40)

Claim 16
The method of claim 1, wherein the selected node layout can be debugged visually (VB, page 782, Debugger)

Claim 17
The method of claim 1, wherein the parameter and properties values of the nodes can be changed dynamically based on business rules (SW, col 3, line 65 to col 4, line 34 and claim 11).

Claim 18
The method claim 1, wherein the parameter and properties values can be linked to variables (SW, col 7, line 65 to col 8 64 – properties and col 36, lines 20 – 30).

Claim 19
The method of claim 1, wherein the application logic is directly executed without compilation of application logic. (SW, col 2, lines 5 – 16, DLL)

Claim 20
The method of claim 1, wherein the application logic can be paused and saved during execution (SW, Figure 16, Wait and col35, lines 7 – 20 and WFT, page 8-19).

Claim 21
The method of claim 21, wherein the saved application logic can be restored and resumed (SW, col 35, lines 50 – 65, wait and cancel)

Claim 22
The method of claim 22, wherein the saved application logic can be restored and execution resumed on a copy of the server program on a computer other than where it was initially started (See cancel in claim 21 and server of claim 1)

Claim 23
SW teaches a method for creating software (SW, Abstract) , comprising: providing a plurality of nodes (See the rejection for claim 1) and a directory of applications (See the rejection for claim 1), each of an application being created by use of at least a portion of the plurality of the nodes(See the rejection for claim 1); selecting at least a portion of the plurality of nodes to create a selected node layout that represent a plurality of application logics(See the rejection for claim 1); defining the application logic by selecting at least one of GUI parameters and options in each selected node (SW, col 24, line 42 to col 25, line 37); executing the selected node layout by a server program (VB, Server, pages 547 – 549). visually showing the selected node layout as a visual node layout ; monitoring a flow of control through each node in the node layout during execution by showing individual node execution measurement; and wherein the individual node execution measurements include usage totals, whole execution time and average execution time. See the rejection for claim 1.
using a solution engine that utilizes networked objects to track, control and manage user sessions (See Thompson teaches Figure #27, Users functionality, #402 and #404) , wherein the solution engine interfaces with an observation (Thompson, Figure 5, Comment Window) , personalization (Thompson, col 8, lines 60-62) and pricing engine as directed by specific nodes (Thompson, col 8, lines 8 – 25 and col 18, lines 41-56).

SoftWire (SW) teaches the use of icons to perform object based programming. The icons generate Visual Basic code. Visual Basic (VB) is a popular object programming language which enables add-ins/-ons such as SoftWire (VB, Chapter 3 – VB6 Template Manager, page 62) features such as web development and client server architecture. Softwire is an add-on to Visual Basic which teach a rapid development environment. Template teaches the ability to drill down within nodes and view the underlying performance information (WFT, Chapters 8 and 9) of the workflow model both in a simulation (Chapter 8) and the deployed system (Chapter 9). Thompson teaches automated configuration (Thompson, Abstract)Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of rapid deployment system of SW and VB with the node statistical abilities of Template with the ability to perform automated configuration, because the ability to determine the performance of a workflow systems yields information on how to tune the system and automated configuration yield information on how to tune and calibrates systems based on usage.

Claim 27

The method of claim 23, further comprising: providing documentation of a functional use of a node. (VB, page 737 – add in documentation to Help)

Claim 28
The method of claim 23, further comprising: providing a graphic description of a plurality of nodes that represent a full application logic (SW, col 1, line 47 – col 2, line 16).

Claim 29
The method of claim 23, further comprising: creating a history of different versions of the application logic (VB, page 72, version).

Claim 30
The method of claim 23, further comprising: creating access control of the application logic. (VB, page 544, getters controlled by PUBLIC or PRIVATE see pages 218 – 219 and SW, col 31, lines 22-24).

Claim 31
The method of claim 30, wherein the access control provides single access of the application logic for purposes of modification and multiple access of the application logic for purposes of viewing. (See the rejection for claim 30 – controlled access by PUBLIC or PRIVATE see pages 218 – 219 and SW, col 31, lines 22-24).

Claim 32
The method of claim 23, further comprising: automatically validating the application logic against errors. (SW, Col 1, lines 42-46 , col 2, lines 10 – 16).

Claim 33
The method of claim 23, further comprising: aggregation at least a portion of the plurality of nodes to create an aggregated node. (SW, col 6, lines 38 – 46 and col 7, lines 7 – 12 – Also inherent in Object technology for extensible programming - aggregation is a form of inheritance)

Claim 34
The method of claim 33, wherein the aggregated node is an application logic. (See claim 33 – methods are inherited).

Claim 35
The method of claim 34, wherein the aggregated node can be used different application logics. (SW, Figure 15 under General shows the reusable components – Reuse is one of the common benefits of Object Technology, SW, figures 13 – 15).

Claim 36
SW teaches a method for creating software (SW, Abstract), comprising: providing a plurality of nodes (See the rejection for claim 1) and a directory of applications (See the rejection for claim 1), each of an application being created by use of at least a portion of the plurality of the

nodes (See the rejection for claim 1) ; selecting at least a portion of the plurality of nodes to create a selected node layout that represent a plurality of application logics nodes (See the rejection for claim 1); defining external application interfaces nodes (SW, Figure 4C, Data Acquisition and Claim 5 external connections for email and Internet) ; executing the selected node layout by a server program nodes (VB, Server, pages 547 – 549).
visually showing the selected node layout as a visual node layout ; monitoring a flow of control through each node in the node layout during execution by showing individual node execution measurement; and wherein the individual node execution measurements include usage totals, whole execution time and average execution time (see the rejection for claim 1.
using networked objects to track, control and manage user sessions (See Thompson teaches Figure #27, Users functionality, #402 and #404) to interface with an observation (Thompson, Figure 5, Comment Window), personalization (Thompson, col 8, lines 60-62) and pricing engine as directed by specific nodes (Thompson, col 8, lines 8 – 25 and col 18, lines 41-56).

SoftWire (SW) teaches the use of icons to perform object based programming. The icons generate Visual Basic code. Visual Basic (VB) is a popular object programming language which enables add-ins/-ons such as SoftWire (VB, Chapter 3 – VB6 Template Manager, page 62) features such as web development and client server architecture. Softwire is an add-on to Visual Basic which teach a rapid development environment. Template teaches the ability to drill down within nodes and view the underlying performance information (WFT, Chapters 8 and 9) of the workflow model both in a simulation (Chapter 8) and the deployed system (Chapter 9). Thompson teaches automated configuration (Thompson, Abstract)Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of rapid deployment system of SW and VB with the node statistical abilities of Template with the ability to perform automated configuration, because the ability to determine the performance of a workflow systems yields information on how to tune the system and automated configuration yield information on how to tune and calibrates systems based on usage.

Claim 37
The method of claim 36, further comprising: establishing conditions for execution of the selected node layout. (SW, col 36, lines 21 – 41).

Claim 38
The method of claim 37, wherein the conditions for the execution include time based events. (SW, col 30 , lines 49 – 65).

Claim 39
The method of claim 37, wherein the conditions for the execution include programmatic events. (SW, col 27 line 40 to col 28, line 15).

Claim 40
The method of claim 39, wherein selected programmatic events create a trigger for the execution of the selected node layout. (SW, Claim 39 and col 28, lines 15 – 64).

Response to Arguments

7. Applicant's arguments with respect to claims 1 – 23, 27 - 40 have been considered but are moot in view of the new ground(s) of rejection.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Todd Ingberg/
Primary Examiner

TI